International Journal of Cybersecurity Engineering and Innovation

ARTICLE INFO

Article History: Received: 05-01-2026, Revised: 20-02-2026, Accepted: 01-03-2026, Published: 04-03-2026
Corresponding author Email: Mahmood.alshareedah@stu.edu.iq
DOI:

This is an open access article under the CC BY 4.0 license
(http://creativecommons.org/licenses/by/4.0/).
Published by ITAP Publisher.

Vol. 2026 No.1

Real-Time Spectrum Sensing on an RTL-SDR-Based IoT Platform

Abdullah Hussein Talib[1], Aseel Hameed AL-Nakkash[1], Ahmed Ghanim Wadday[2], Ali A. Abed[3] and Mahmood A. Al-Shareeda[4*]

[1] *Computer Techniques Engineering, Middle Technical University, Iraq*

[2] *Al-Furat Al-Awsat Technical University, Iraq*

[3] *Mechatronics Engineering Department, University of Basrah, Basrah, Iraq*

[4] *Southern Technical University and Al-Ayen University, Iraq*

**Corresponding author:* Mahmood.alshareedah@stu.edu.iq

Abstract

This focuses on the application of cognitive radio (CR) technologies in Internet of Things (IoT) networks for dynamic spectrum access among a large number of IoT end nodes. Spectrum Sensing (SS) is a fundamental function in CR-IoT nodes, which allows cognitive devices to sense and identify spectrum holes. In this research, we realize a real-time implementation of SS based on SDRs and ED. Furthermore, we use Machine Learning (ML) tools, specifically Support Vector Machine (SVM), to develop an offline sample of energy detection. We use USRP N210 and RTL-SDRs as hardware in the laboratory's unblended test bed for the SS methodology. The results obtained have shown that energy detection is the promising method at SNR -10 dB and higher. This is because of the difficulty in accurately choosing λ when SNR < -10 dB. The investigation also used SVM with an off-line training of a set of energy detection samples to estimate status of the channel. Simulation reveals however that energy detection is limited in low SNR regimes, notably in the selection of an adequate threshold. Alternatively, we have introduced a novel machine learning technique using SVM that can operate more effectively than the existing ones for spectrum sensing in harsh environment. It is successful due to SVM's capability to accommodate complicated decision boundaries and learns from samples.

Keywords: Cognitive Radio (CR), Spectrum sensing, Real-time, Internet of Things (IoT) networks, Machine Learning (ML).

1. Introduction

The IoT primary purpose is to make internet connectivity available to a vast number of devices or "things" via wireless communications. The scarcity of available frequency spectrum raises an obstacle for providing wireless connectivity to a wide range of devices. How to solve this problem would be as follows: dynamic spectrum access with intelligent spectrum management is required. CR technology has been considered as an effective solution for solving the problem of spectrum scarcity and unpredictable characteristics in wireless networks [1]. In CR, SS is an important operation to sense and utilize a spectrum holes efficiently. At present, the spectrum assigned to wireless systems is static and managed by government regulation. Today, the licensing of radio frequency (RF) spectrum is allocated in large, long-term blocks to a number of entities that may range from companies to individuals. Nevertheless, pervasive devices in the IoT scenario will lead a very high frequency spectrum demand in particular due to the growing number of wireless devices. In the light of spectrum scarcity, existing static frequency assignment schemes are not suitable for the increasing number of high rate wireless devices. Most of spectrum is sitting around unused as well. One such study in 2004 found that only about 13% of available spectrum was used in New York City when demand for bandwidth was greatest.

In CR setting, the RF spectrum is largely reserved for authorized holders known as Primary Users (PUs), who hold precedence right of access to the spectrum. Meanwhile, CR users or secondary users (SUs) can continually keep track of the transmission behavior of Pus that are operating in a certain frequency band, with respect to both space and time factor. The aim of Sus is to reuse the spectrum while this is in an idle state for PUs. This infrastructure constitutes the basis of an overlay CR network.

Alternatively, in some cases, PUs and SUs can interoperate simultaneously on a Single frequency band. However, for the correct functioning and to limit aggressive interferences due to secondary's transmissions it is necessary that the total power transmitted by SU has to be low enough. This requirement is essential in order to restrict the total interference power level under the permissible constraint according to [2]. SS is an important component in the design of CR systems and has received extensive attention from academia. In addition to SS, the other major research issues CR systems include spectrum sharing spectrum allocation and management, SU transmission [3]. One of these parts is the SS algorithm that it is used to scan one frequency band and find out where PUs are existed or occupied. The reliability and accuracy of the SS process affect such performance of the CR system.

In an overlay CR network, SUs faces the challenge of performing reliable SS to maximize their throughput without interfering with PUs. SS performance is typically evaluated using two crucial measures: the probability of detection (P_d) and the probability of false alarm (P_f). Different approaches have been suggested for SS [3], [4], including energy detection, matched filtering, cyclostationary feature identification, and generalized likelihood ratio test.

Recent research has concentrated on real-time hardware implementations and verifications of CR algorithms. In [5] authors used real-world energy detection measurements with USRP (Universal Software Radio Peripheral) and GNU Radio. Another study in [6] used low-cost software-defined radio (SDR) hardware to do coordinated and distributed wideband SS over a geographic area, with energy detection as the sensing approach. The findings of these studies were compared to those obtained utilizing specialized professional equipment, such as the CRFS RFeye Node. Furthermore, authors in [7] created and deployed a CR platform based on USRP, which allows for the detection of spectrum gaps and access to the most appropriate frequency band for communication needs. Authors in [8] proposed a SS algorithm called Deep Learning Based Spectrum Sensing (DBSS). Instead of standard energy thresholding approaches, this algorithm employs deep learning techniques, notably a Convolutional Neural Network (CNN) as a detector. The suggested method comprises training the CNN model using actual spectrum data to increase SS performance. Data in an experimental signal transmission and receiving scenario facilitated by software-defined radio (SDR).

The main contributions of this manuscript are as follow: Section 2 reviews the Energy detection based SS and elaborates on its principles and methodology. In section 2, ML is presented and specifically SVM as a method to SS. Section 3 introduces the experimental setup, made of RTL-SDR and GNU Radio. Results of the experiments are shown and discussed in Section 4). We conclude the paper in Section 5 by summarizing our results and discussing their implication.

2. Research Background

2.1 Energy Detector

Energy Detection (ED) based SS is a method that is frequently used due to its low computational complexity and ease of implementation. This technique is widely used because it does not require prior knowledge of the PU signals. In ED, a Threshold (λ) that is often reliant on the noise floor is used to compare the output of the energy detector with to determine the existence of a main user. However, there are several challenges associated with this method. For instance, selecting an appropriate λ can be challenging. Additionally, energy detection struggles to distinguish between interference, PU signals, and noise, SS techniques aim to accurately detect the presence of PU signals. However, these techniques may experience reduced performance in scenarios characterized by low signal-to-noise ratio (SNR) conditions [9]. Consider the received signal denoted as

$$y(k) = s(k) + n(k)\,, k = 1\ldots\ldots K \quad (1)$$

Where $s(k)$ is the signal, $n(k)$ is noise and k present number of samples. The energy detection decision metric is

$$Z = \sum_{k=0}^{K} |y(k)|^2 \quad (2)$$

The decision metric, denoted as $Z = 1/K \sum_{k=0}^{K} |Y(k)|^2$, represents the signal in the frequency domain, where k is the length of the observation vector. Where $|Y(k)|^2$ represents the K-point Discrete Fourier Transform (DFT) of $y(k)$.In the measurement setup described in the following section, the frequency domain counterpart of this equation will be utilized. SS is the process of assessing whether or not a sent signal is using the observed spectrum. It is possible to think of this as a binary hypothesis testing problem since the region of the spectrum being examined is either occupied (H_1) or vacant (H_0) as equation (3).

$$\begin{cases} H_1: y(k) = s(k) + n(k) \\ H_0: y(k) = n(k) \end{cases} \quad (3)$$

The P_d and the P_f are two performance measures that may be used to evaluate a detection algorithm's efficacy. The likelihood of accurately identifying a signal that is truly present is represented byP_d. P_f, on the other hand, represents the likelihood of mis indicating a signal's existence when it isn't. P_f, on the other hand, describes the likelihood of mistakenly detecting a PU signal that isn't truly there. The definition of P_d and P_f is [9].

$$P_d = P_r(Z > \lambda | H_1) \quad (4)$$

$$P_f = P_r(Z > \lambda | H_0) \quad (5)$$

The selection of the λ parameter plays a crucial role in determining the values of the two parameters, P_d and P_f. A lower value of P_d indicates underutilization of the spectrum and it is desirable to keep it as high as possible. Lowering the λ is one way to get a high P_d, however doing so also results in more false alarms. These definitions allow us to define the chance of missing a signal P_m as follows:

$$P_m = 1 - P_d \quad (6)$$

2.2 Machine Learning Techniques

In this section, we investigate techniques that can be employed without any prior knowledge of the channel. However, these techniques require cooperation from the primary user, who is responsible for providing the fusion center with labeled samples corresponding to previous sensing periods. The training process involves an automated approach to approximate a function that maps the estimated energy samples from the secondary users to the labels provided by the primary user. Once the training phase is complete, the models are capable of determining the channel status based on unseen energy samples.

2.2.1 Support Vector Machine

The SVM is a maximum-margin classifier. In simple terms, the SVM constructs a hypothesis function that produces a positive output for hypothesis H1 and a negative output for H0. The SVM aims to find the decision boundary that maximizes the margin between different classes, allowing for effective classification. Figure 1 Visual representation of a decision plane in a SVM. The optimal hyperplane serves as a separator between two classes, represented by filled circles and empty circles. It is positioned at the midpoint of the maximum margin defined by the support vectors, which are the closest data points from each class to the decision boundary.

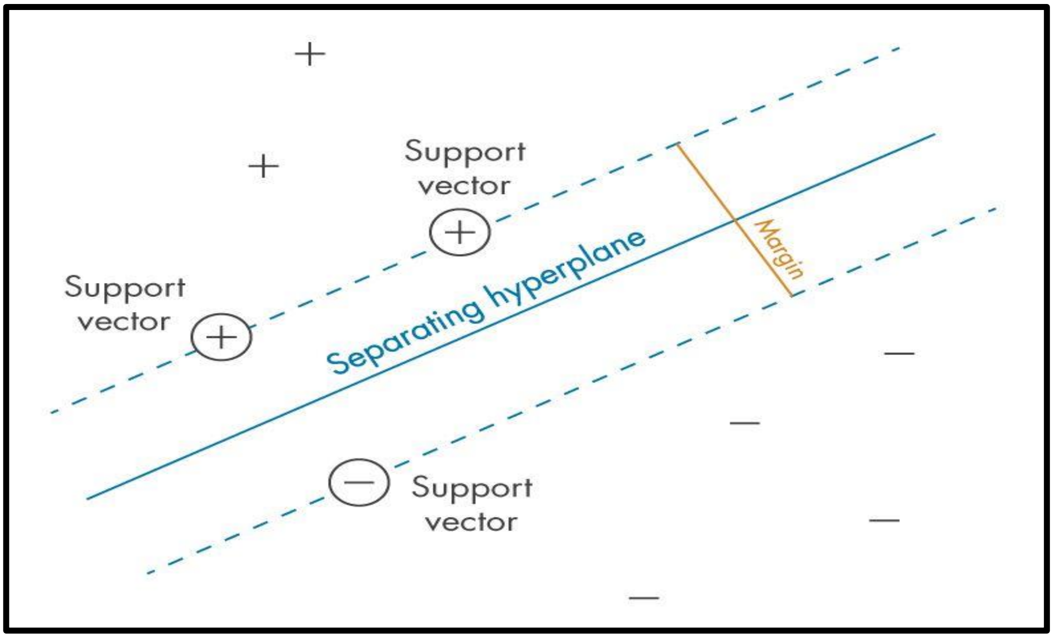

Figure 1. Support vector machine (SVM)

3. Research Methodology

3.1 Set up of Measurements

3.1.1 RTL SDR

The IoT node comprises a computer and an SDR dongle, specifically the R828D-based RTL SDR platform. The RTL SDR platform comprises an 8-bit analog-to-digital converter (ADC) and a USB data pump. The functionality of the RTL SDR can be understood through the behavioral level model depicted in Figure 2.

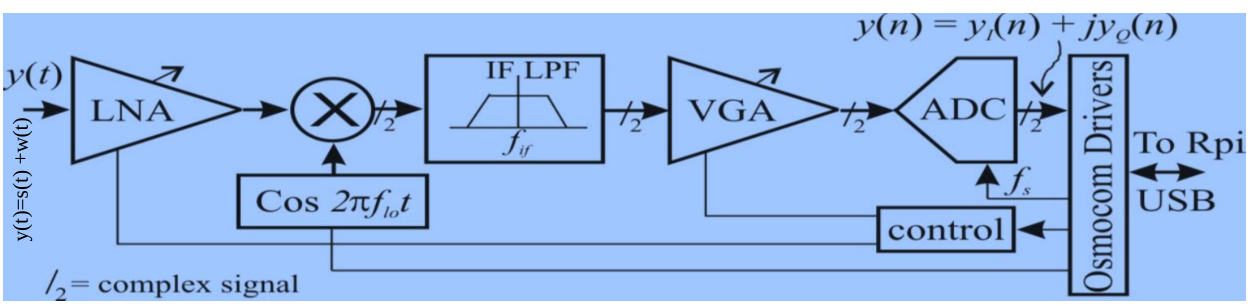

Figure 2. Block diagram of RTL SDR.

The intended signal $s(t)$ and background noise$W(t)$, generated by the receiver front end and antenna, make up the input signal $y(t)$ in this arrangement. A wideband low-noise amplifier (LNA) is used to amplify the RF signal before it is downconverted to the lower intermediate frequency (IF), abbreviated asf_{if} . The incoming signal is multiplied with a locally produced carrier signal of frequency f_{lo} to achieve downconversion, and the resultant product is filtered using a filter tuned at frequency f_{if} . A variable gain amplifier (VGA) with automatic gain control (AGC) is used to further amplify the IF signal. After that, it is sampled f_s times per second. Through a USB connection, the sampled and quantized signals may be accessed by a computer for additional processing and analysis.

3.2 Spectrum Sensing using GNU Radio

This SDR signal received with RTL SDR is processed on the PC using GNU Radio software framework. Open source software toolkit GNU Radio for SDR development where all baseband signal processing is performed by the software program. It can produce various modulation waveforms, demodulate signals, filter processing and signal synthesis through filters; and all other kinds of signal advancing process. The working SDR system may be built using the library of signal-processing blocks provided by the GNU Radio tools. The flow graphs that describe the signal processing chain are built using Python, and these blocks are implemented in C++ [10]. The graphical user interface (GUI) for GNU Radio is called GNU Radio Companion (GRC), which enables users to create GNU Radio applications in a way like to MATLAB Simulink or National Instruments LabVIEW. With GRC, a "drag and drop" approach makes it simple to design blocks and connections. To create the appropriate signal processing flow, GRC's blocks may be joined to one another and each block has a variety of functions for connections to be successful, the inputs and outputs of the blocks as well as the data types for the inputs and outputs must match [10]. It is possible to use a GRC implementation in the context of energy detection SS. Figures 3 and 4 show an example of this being done. Although in actuality it is not necessary to be aware of the precise configuration (modulation, bandwidth, data rate, etc.) of the main user (PU), an example implementation of a PU is depicted in Figure 3.

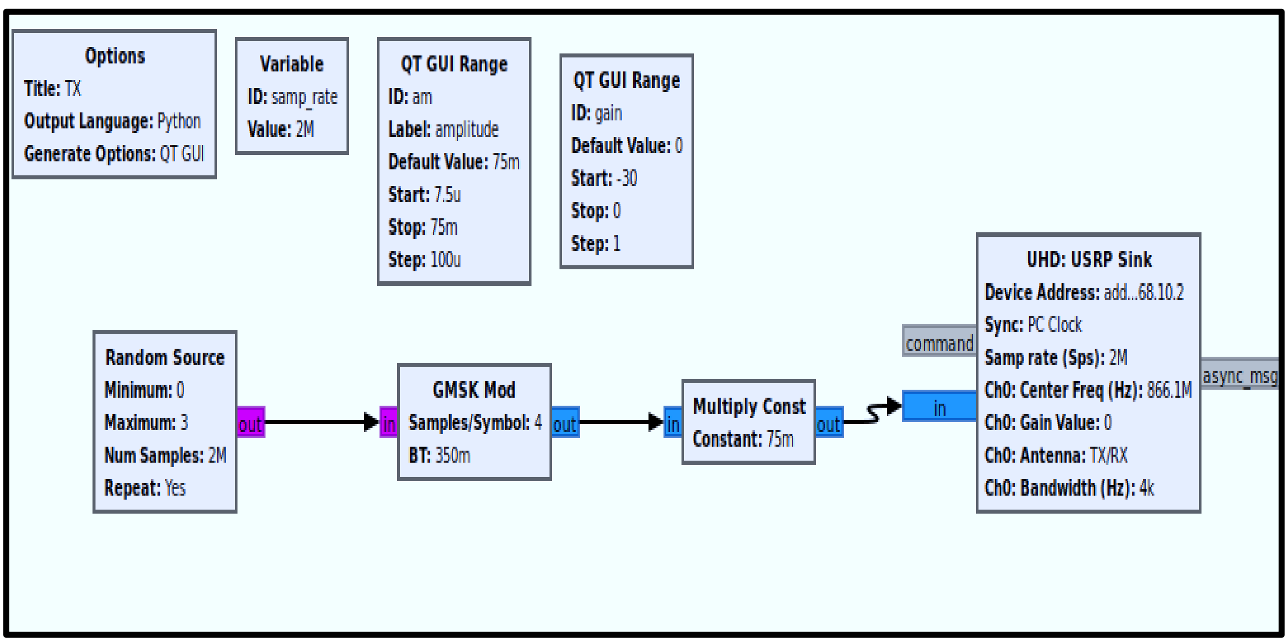

Figure 3. Block diagram of a transmitter running on USRP using GNU Radio.

Using the Universal Software Radio Peripheral (USRP) SDR, the primary source broadcasts random data that has been modulated using Gaussian minimum shift keying (GMSK). Figure 4 depicts the sensing setup located in SU. It works by applying the Parseval theorem on equation (2), yielding

$$M = \frac{1}{N}\sum_{n=0}^{N}|Y(n)|^2 \tag{8}$$

Where $Y(n)$ is the discrete Fourier transform (DFT) or fast Fourier transform (FFT) of $y(n)$ at N points. DFT can be used to calculate power spectrum density $S(n)$ as

$$S(n) = \frac{1}{N}|Y(n)|^2 \tag{9}$$

The 'QT GUI Number Sink' block shows the moving average value of the measured power, computed over number of samples. The SNR of the received signal may be estimated using this approach.

$$SNR = \frac{P_s}{P_w} = \frac{P_t}{P_w} - 1 \tag{10}$$

Let P_W stand for the signal power determined without the PU. The signal power P_t is the total of the signal power P_s and the noise power P_w when it is measured in the presence of the PU. As seen in Figure 4, the data flow processing through the CR- IoT node or SU may be summarized as follows:

1. The RTL SDR source intercepts a signal with a bandwidth of f_{BW}, centered around f_c (carrier frequency).
2. The output of the RTL SDR source is a continuous stream of eight-bit symbols, which is then converted into vectors of length 1024.
3. A 1024-point Fast Fourier Transform (FFT) is applied to each of these vectors to calculate the discrete Fourier transform (DFT).
4. The squared magnitude $|Y(n)|^2$ is calculated for each DFT output.
5. The Power Spectral Density (PSD) $S(n)$ is obtained by multiplying the squared magnitude with 1/N, where N is the length of the DFT.
6. The calculated values of PSD are saved for later processing and the calculation of the decision metric M using MATLAB.

This data flow processing allows for spectral analysis of the received signal, enabling further analysis and decision-making based on the obtained PSD values. The MATLAB software is used for subsequent processing steps, such as decision metric calculation, which helps determine the presence or absence of the PU signal.

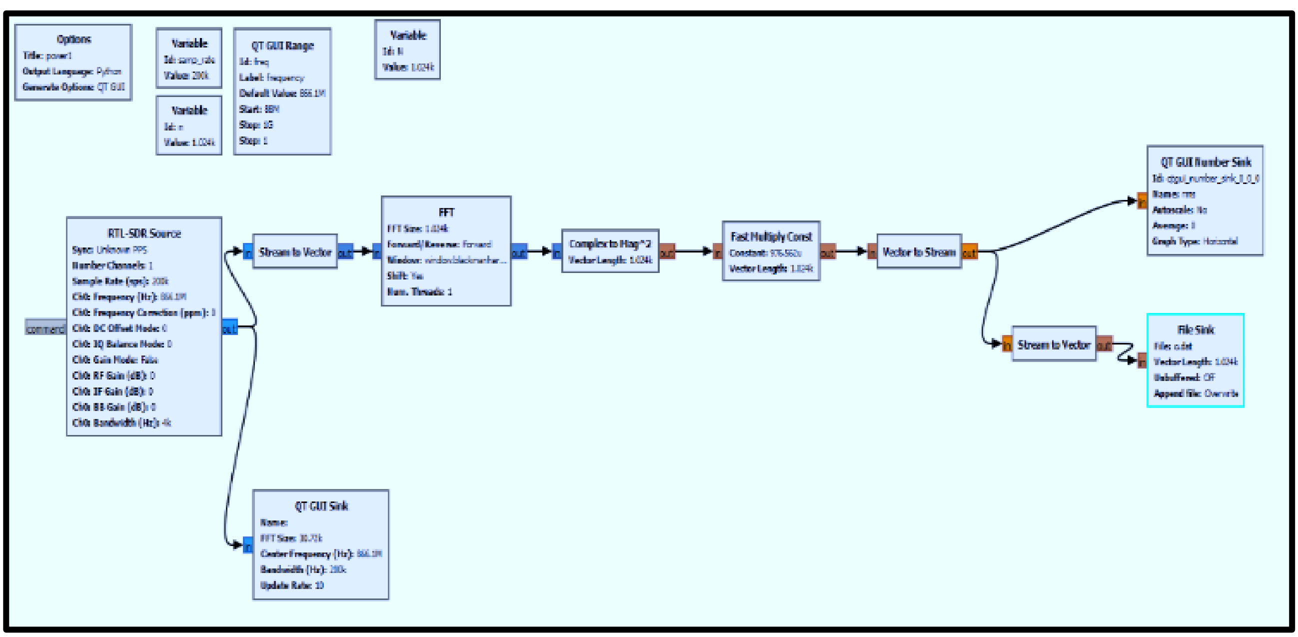

Figure 4. GNU Radio block diagram of receiver implemented on RTL SDR.

4. Findings of the Study

The GNU radio environment is operated on the PU node's Ubuntu Linux xxxx LTS operating system. The transmitter used by GRC is seen on Figure 3. The symbols are sent at center frequency fc = 866.1 MHz using GMSK (gaussian minimum shift keying) modulation. Given the introduction of LoRa devices for Internet of Things applications in the 868 MHz range, this frequency is important. In the 868 MHz LoRa range, which runs from 865.2 MHz to 868 MHz, there are eight channels [11].

RTL SDR and a computer make up the sensing setup, or CR IoT node, as was previously mentioned. Utilizing GNU radio, the system paradigm in Fig. 4 is put into practice. The operating system of the PC is Windows. The center frequency in this configuration is set at fc = 866.1 MHz, and the sampling rate is set to fs = 2×10^5 samples per second. A PC linked to USRP that serves as a PU transmitter and a spectrum sensor system SU made up of RTL SDR and a computer make up a real-time measurement setup as shown in Figure (5). Figures (6) and (7) show, respectively, screenshots of the received signal's spectrum in the absence and presence of a signal from PU.

Figure 5. An IoT node consisting of RTL-SDR as SU and USRP210N as PU.

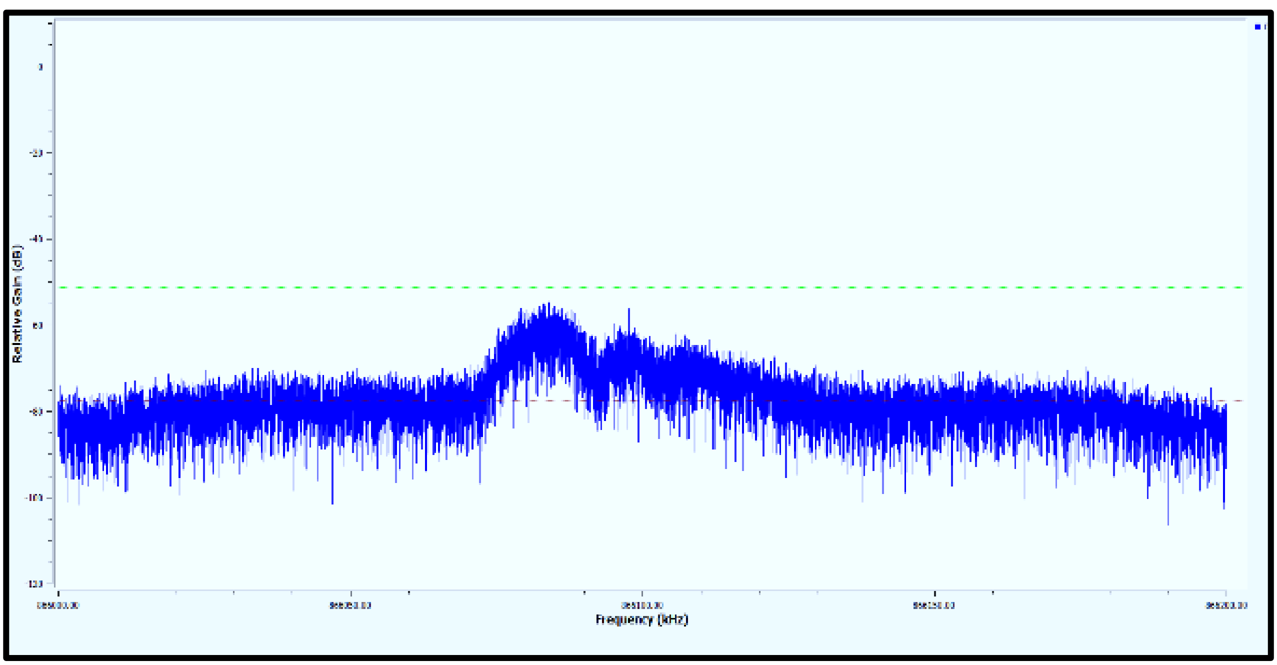

Figure 6. Spectrum of the signal that was received in the absence of a PU signal.

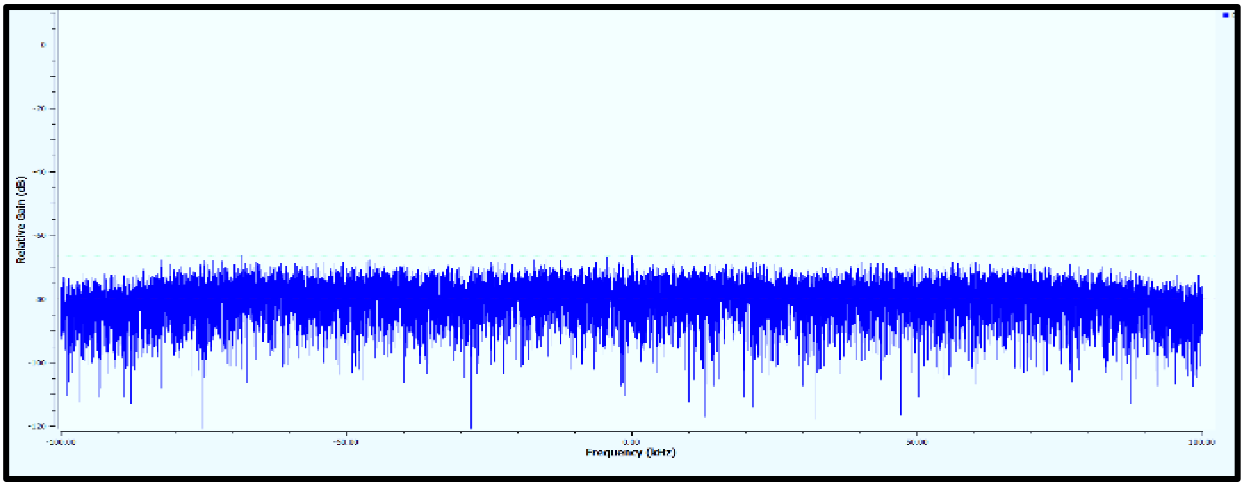

Figure 7. Spectrum of the incoming signal when a PU signal is present.

Figure 8 depicts the measurement results for the P_f, with three curves representing different values of λ (-30.42 dB, -30.45 dB, and -30.43 dB). From this figure, it is evident that when the SNR exceeds -15 dB, the performance remains consistent across all λ values. Therefore, selecting an appropriate λ becomes relatively straightforward in such cases. However, when the SNR falls below -15 dB, any adjustment to λ, no matter how minor, can significantly impact the performance of the system.

The energy detection performance is shown in Fig. 8 and Fig. 9. The choice of the λ factor in energy detection is very important and it has a big impact on the decision about two basic parameters and Pf. A small indicates that not enough information is provided by the spectrum; it is desired to have Pd as high as possible. It is possible to set a high by adjusting the value of λ. This also results in an escalating, meaning a growing possibility of false alarms. Therefore, a balance between and needs to be made in energy detection.

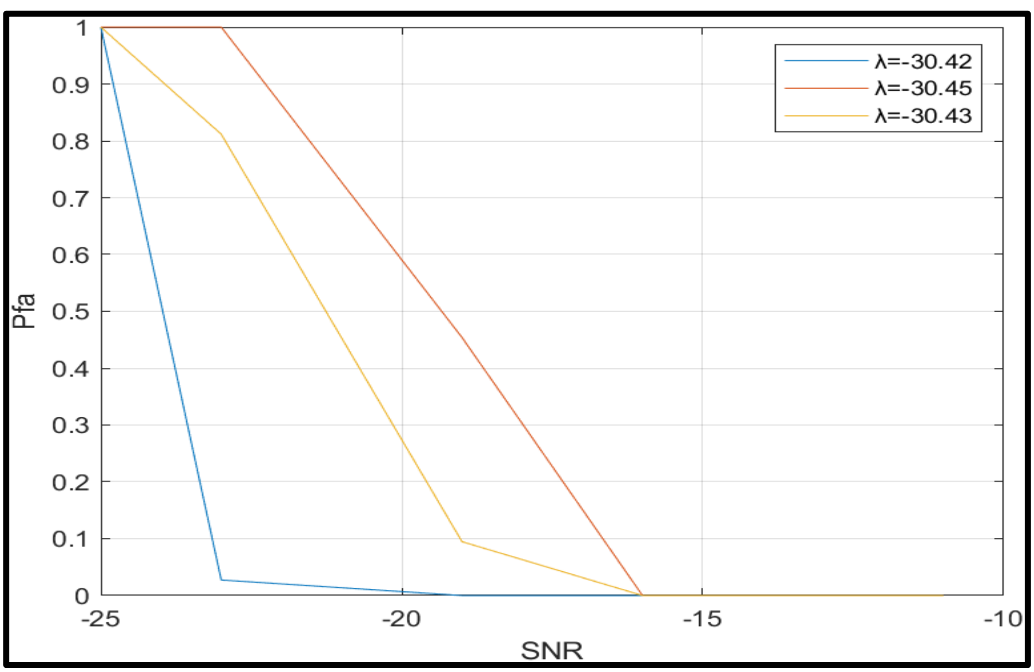

Figure 8. Pfa vs SNR

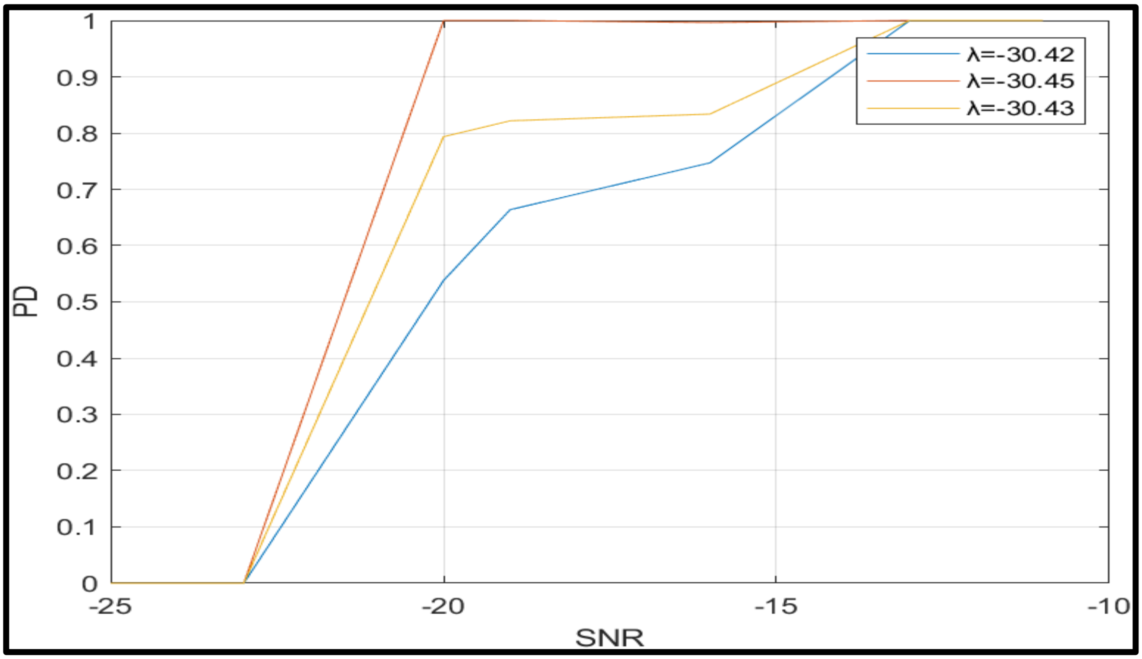

Figure 9. Pd vs SNR

Where it is to be noted from Figures 10 and 11 that confusion matrix between the SVM classifier in varying SNR ranges (-25 dB & -10 dB) and easy, medium. If we look at the confusion matrix, we can verify that our is 93.1% and the % is =11%. These results were obtained through offline SVM trained with real data. In our scheme, since we use the real data to train SVM, it can well learn the intrinsic patterns and features during online applications even in diverse SNR levels. It can be also observed that, SVM has performed reasonably well even without the cost of comparison with λ and thus it emphasizes yet again on the efficiency of ML approaches over traditional energy detection techniques. It can make SVM more practical through real data training for the machine.

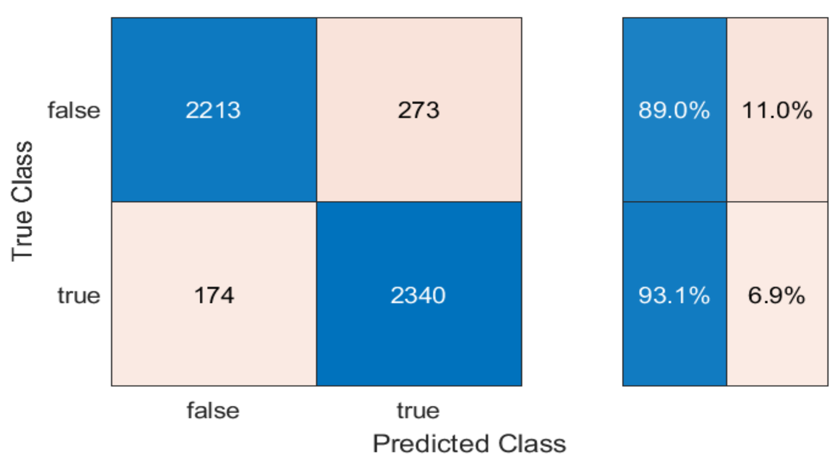

Figure 10. Confusion matrix for GSVM

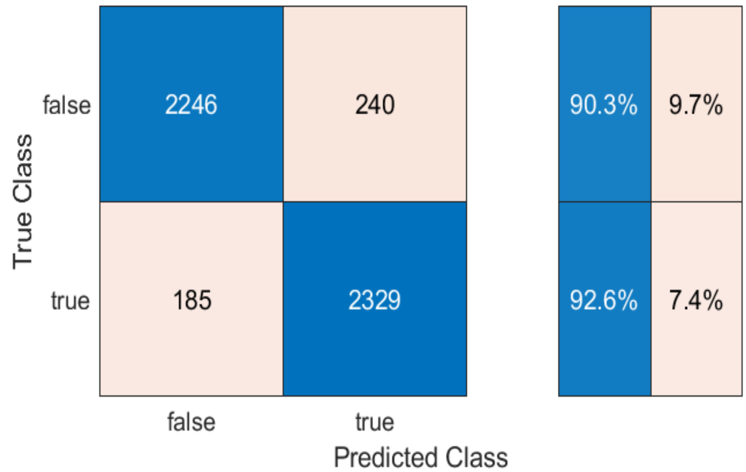

Figure 11. Confusion matrix for LSVM

5. Conclusion

The Experimental Configuration of Energy Detection SS The experiment setup for energy detection spectra sensing is presented in the article. The setup was developed in a RTL SDR on the personal computer, it was tailored to IoT Cognitive Radio (CR) applications. The software was implemented using GNU Radio on Windows. The results obtained have shown that energy detection is the promising method at SNR -10 dB and higher. This is because of the difficulty in accurately choosing λ when SNR < -10 dB. The investigation also used SVM with an off-line training of a set of energy detection samples to estimate status of the channel. Simulation reveals however that energy detection is limited in low SNR regimes, notably in the selection of an adequate threshold. Alternatively, we have introduced a novel machine learning technique using SVM that can operate more effectively than the existing ones for spectrum sensing in harsh environment. It is successful due to SVM's capability to accommodate complicated decision boundaries and learns from samples.

References

[1] Ghasemi, A., & Sousa, E. S. (2008). Spectrum sensing in cognitive radio networks: Requirements, challenges and design trade-offs. *IEEE Communications Magazine, 46*(4), 32–39. https://doi.org/10.1109/MCOM.2008.4481338

[2] Biglieri, E., Goldsmith, A. J., Greenstein, L. J., Mandayam, N. B., & Poor, H. V. (2013). *Principles of cognitive radio*. Cambridge University Press.

[3] Sadhukhan, P., Kumar, N., & Bhatnagar, M. R. (2013). Improved energy detector based spectrum sensing for cognitive radio: An experimental study. In *Proceedings of the 2013 IEEE Annual India Conference (INDICON)*. https://doi.org/10.1109/INDCON.2013.6725925

[4] Subhedar, M., & Birajdar, G. (2011). Spectrum sensing techniques in cognitive radio networks: A survey. *International Journal of Next-Generation Networks, 3*(2), 37–51. https://doi.org/10.5121/ijngn.2011.3203

[5] Yusof, K. H., Rahim, M. K. A., Yusof, K. M., & Jalil, M. E. (2015). Energy detection spectrum sensing measurement using GNU radio and USRP B200 at Wi-Fi frequency. *Lecture Notes in Electrical Engineering, 344*, 223–230. https://doi.org/10.1007/978-3-319-17269-9_24

[6] Grönroos, S., Nybom, K., Björkqvist, J., Hallio, J., Auranen, J., & Ekman, R. (2016). Distributed spectrum sensing using low cost hardware. *Journal of Signal Processing Systems, 83*(1), 5–17. https://doi.org/10.1007/S11265-015-1033-1

[7] Huang, Z., Wang, W., & Zhang, Y. (2012). Design and implementation of cognitive radio hardware platform based on USRP. In *IET Conference Publications* (Vol. 2011, No. 586 CP, pp. 160–164). https://doi.org/10.1049/CP.2011.0651

[8] Hussein, A. T., Kivanc, D., Abdullah, H., & Falih, M. S. (2023). Deep learning based spectrum sensing method for cognitive radio system (pp. 319–331). https://doi.org/10.1007/978-3-031-27099-4_25

[9] Yücek, T., & Arslan, H. (2009). A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys & Tutorials, 11*(1), 116–130. https://doi.org/10.1109/SURV.2009.090109

[10] Gandhiraj, R., & Soman, K. P. (2014). Modern analog and digital communication systems development using GNU Radio with USRP. *Telecommunication Systems*, *56*(3), 367–381. https://doi.org/10.1007/S11235-013-9850-7

[11] Bharadwaj, A. S., Rego, R., & Chowdhury, A. (2017). IoT based solid waste management system: A conceptual approach with an architectural solution as a smart city application. In *Proceedings of the 2016 IEEE Annual India Conference (INDICON)*. https://doi.org/10.1109/INDICON.2016.7839147